



# Instituição de Tecnologias Avançadas do Porto

## Trabalho Final da Disciplina de DAS

Implementação de Gestão de Repositórios com GitFlow

**Gustavo Lopes e Rui Fangueiro**  
Curso: Desenvolvimento de Software

March 7, 2025

# Contents

<b>1</b>	<b>Código Git Utilizado</b>	<b>2</b>
<b>2</b>	<b>Link de Acesso ao Repositório</b>	<b>2</b>
<b>3</b>	<b>Operações Efetuadas no Git</b>	<b>2</b>
3.1	1. Carregamento Inicial do Ficheiro no Branch <code>develop</code> . . . . .	2
3.2	2. Cinco Alterações no Branch <code>develop</code> . . . . .	2
3.3	3. Release e Contribuição ao Branch <code>master</code> . . . . .	3
3.4	4. Hotfix Realizado . . . . .	3
<b>4</b>	<b>Objetivos do Projeto</b>	<b>3</b>
4.1	Objetivo Geral . . . . .	3
4.2	Objetivos Específicos . . . . .	3
<b>5</b>	<b>Justificativa</b>	<b>4</b>
<b>6</b>	<b>Metodologia</b>	<b>4</b>
6.1	Ferramentas Utilizadas . . . . .	5
<b>7</b>	<b>Resultados Esperados</b>	<b>5</b>
<b>8</b>	<b>Considerações Finais</b>	<b>5</b>

# 1 Código Git Utilizado

O código Git utilizado para a criação do repositório pode ser visualizado através do link do repositório no GitHub. O repositório foi configurado com a metodologia GitFlow, permitindo uma gestão clara das versões, branches e alterações de código.

Para clonar o repositório localmente, utilize o seguinte comando:

```
git clone https://github.com/GustavoMoraisLopes/FinalProjectDASgr.git
```

## 2 Link de Acesso ao Repositório

Para acessar o repositório completo e acompanhar o progresso do projeto, utilize o seguinte link: <https://github.com/GustavoMoraisLopes/FinalProjectDASgr>.

## 3 Operações Efetuadas no Git

Ao longo do desenvolvimento do projeto, foram realizadas várias operações no repositório Git. Abaixo, descrevo as operações mais relevantes que foram efetuadas com base no modelo GitFlow.

### 3.1 1. Carregamento Inicial do Ficheiro no Branch develop

A primeira etapa consistiu em criar o repositório e adicionar o arquivo inicial do relatório, que foi colocado no branch **develop**. Este branch é utilizado para desenvolvimento e será o local onde todas as alterações principais ocorrem antes de serem integradas ao branch **master**.

- Criação do repositório e configuração inicial.
- Adição do arquivo `relatorio.md` com a introdução do projeto.
- Commit inicial com o nome `Carregamento inicial do relatório`.

### 3.2 2. Cinco Alterações no Branch develop

Ao longo do desenvolvimento, foram feitas cinco alterações significativas no conteúdo do relatório. Cada uma dessas alterações foi registrada com um commit no branch **develop**, permitindo um controle eficaz das modificações.

- **Alteração 1:** Adição da seção "Objetivos do Projeto", detalhando os objetivos gerais e específicos.
- **Alteração 2:** Inclusão de uma revisão de literatura, citando as principais fontes e referências utilizadas.
- **Alteração 3:** Explicação detalhada sobre a metodologia aplicada, incluindo o uso do GitFlow para controle de versões.
- **Alteração 4:** Introdução da seção "Resultados Esperados", especificando os objetivos finais do projeto.

- **Alteração 5:** Revisão e melhorias no texto, garantindo maior clareza e coesão nas seções anteriores.

Essas alterações foram registradas com commits claros, garantindo que todas as mudanças fossem devidamente documentadas no histórico do Git.

### 3.3 3. Release e Contribuição ao Branch master

Após a conclusão das alterações no branch `develop`, um `release` foi feito, contribuindo para o branch `master`. O branch `master` contém a versão estável e final do projeto. O `release` também marca o momento em que a versão final do relatório foi consolidada.

A operação foi feita através do comando GitFlow `git flow release finish`, e a versão final foi marcada com a tag `v1.0.0`.

### 3.4 4. Hotfix Realizado

Durante o desenvolvimento, foi necessário realizar um `hotfix` para corrigir pequenos erros de codificação e melhorar a exibição do relatório. O `hotfix` foi feito em um novo branch, a partir do branch `master`, e as correções foram aplicadas antes de serem mescladas de volta ao `master`.

## 4 Objetivos do Projeto

### 4.1 Objetivo Geral

O objetivo principal deste projeto foi implementar um sistema de controle de versão eficaz utilizando o GitFlow para gerenciar as diferentes etapas do desenvolvimento de um relatório de projeto, incluindo a revisão contínua do conteúdo e o controle das versões finais.

### 4.2 Objetivos Específicos

- Estudar e implementar o modelo GitFlow para gestão de versões.
- Garantir que todas as versões do relatório sejam devidamente controladas e documentadas no repositório Git.
- Criar um ambiente de colaboração e revisão contínua para garantir a qualidade do projeto.
- Concluir o projeto com um relatório final organizado e devidamente versionado no GitHub.
- Apresentar resultados claros e mensuráveis com a utilização de GitFlow.

## 5 Justificativa

Este projeto foi desenvolvido como parte do trabalho final do curso, com o objetivo de demonstrar a aplicação de boas práticas no uso de **Git** e **GitFlow**, além de garantir o cumprimento de todos os requisitos definidos pela instituição. A escolha de usar GitFlow no gerenciamento de versões do projeto foi motivada pela necessidade de aplicar uma metodologia que garantisse um controle de versões organizado, especialmente em um contexto onde é fundamental manter um histórico claro de todas as modificações feitas ao longo do desenvolvimento do relatório.

Como se trata de um trabalho final, é essencial garantir que todas as etapas de desenvolvimento e os processos de colaboração sejam documentados de forma precisa e sem erros, e o GitFlow se mostrou a solução mais eficiente para gerenciar essa complexidade. Além disso, foi necessário garantir que: - O relatório fosse continuamente atualizado e melhorado ao longo do tempo, sem a introdução de conflitos. - A versão final do trabalho estivesse claramente definida e documentada, atendendo aos requisitos exigidos pela instituição, como o controle de versões e a organização das mudanças. - Houvesse uma metodologia que permitisse a revisão contínua e a finalização de uma versão estável do relatório para entrega.

Assim, o uso do GitFlow foi uma decisão estratégica para garantir a **qualidade do trabalho final** e a **conformidade com os requisitos da instituição**, assegurando que todas as operações realizadas durante o projeto fossem bem geridas e documentadas.

## 6 Metodologia

A metodologia aplicada foi baseada no modelo GitFlow, que estabelece um processo claro para o controle de versões e a gestão de branches. As etapas seguiram o ciclo básico do GitFlow:

- Criar um **branch develop** para o desenvolvimento contínuo.
- Criar **feature branches** para alterações específicas no relatório.
- Criar **release branches** para preparar a versão final do relatório.
- Criar **hotfix branches** para correções de última hora.

Cada uma dessas etapas foi devidamente implementada com a utilização do GitHub, criando uma documentação precisa sobre as modificações e as versões. Além disso, ao longo do processo, foi necessário realizar **merge** entre os branches para consolidar as alterações, garantindo que não houvesse conflitos.

## 6.1 Ferramentas Utilizadas

Foram utilizadas as seguintes ferramentas e recursos para a implementação do projeto:

- **Git**: Sistema de controle de versão distribuído.
- **GitFlow**: Extensão do Git para facilitar o gerenciamento de branches e versões.
- **GitHub**: Plataforma de hospedagem de código e colaboração em projetos de software.
- **Visual Studio Code**: Editor de código utilizado para o desenvolvimento do conteúdo do relatório.
- **Markdown**: Linguagem de marcação para estruturar o conteúdo do relatório.

## 7 Resultados Esperados

O objetivo final do projeto foi criar um relatório que estivesse bem estruturado, revisado e versionado corretamente. Os resultados esperados incluíam a implementação bem-sucedida do GitFlow no projeto, com a capacidade de controlar as versões de maneira eficiente e documentada. Espera-se também que o repositório seja uma boa referência para o uso do GitFlow em projetos de desenvolvimento de software.

Além disso, o projeto teve como resultado a consolidação de habilidades de **colaboração em equipe**, **gestão de versões** e **uso de ferramentas de desenvolvimento modernas**. O GitFlow demonstrou ser uma ferramenta extremamente útil para a organização de projetos complexos.

## 8 Considerações Finais

O projeto demonstrou a eficácia do uso do GitFlow na gestão de versões e controle de alterações em projetos colaborativos. A utilização do GitHub como plataforma de versionamento e controle de código se mostrou essencial para garantir a integridade e a organização do trabalho. A entrega final do relatório foi realizada com sucesso, e o projeto está pronto para ser finalizado.

Foi possível observar que a implementação de boas práticas de controle de versão, como as proporcionadas pelo GitFlow, é essencial para o sucesso de projetos de software, seja em ambientes colaborativos ou individuais. Além disso, a familiaridade com essas ferramentas prepara o desenvolvedor para o mercado de trabalho, onde o controle de versões é uma habilidade altamente valorizada.